

September 13th, 2013

German Feedback to the Point Paper on "BIBFRAME Use Cases and Requirements"

General remarks

This is our feedback to the "BIBFRAME Use Cases and Requirements" point paper. Our remarks cover the more general estimations that we discussed during the meeting on Wednesday, 2013-09-11 at the German National Library.

We are glad to see this paper published, as it provides a concrete, lively, somewhat refreshing and sometimes even funny approach to BIBFRAME. There are no explicit references to the MARC format, which emphasizes the more forward looking and dynamic aspects of BIBFRAME. The paper triggered some lively and substantial talks, and not each remark is meant to be criticizing. We definitely think that the paper is worth commenting on it in depth.

Comments on section 4. "Requirements and Design Objectives"

In this paper, the division into parts 3 and 4 does make sense. However we think that some important parts of section 4 may get lost, and would deserve a paper on their own. This is particularly true for some parts in section 4 with only a loose relationship to one of the uses case, e.g. for section 4.10, which we can imagine to see again in a paper entitled "BIBFRAME and protocols", "BIBFRAME and technical requirements", or something similar.

Regarding serialization issues, we had some discussion. I'm quoting my colleague Lars Svensson here:

"Comments on BIBFRAME use case 4.7: Required Serialisation

The use case document states in section 4.7 that in order to "support the ease of BIBFRAME a required serialization of RDF/XML is defined as the basis for interchange."

This is a somewhat strange requirement since it de facto specifies that the serialisation is more important than the underlying model and makes sense only if the intention is that systems/interfaces should be able to process the data using XML semantics (as opposed to handling it with RDF semantics and seeing SML only as a serialisation). Considering that it is possible to serialise the same RDF graph in RDF/XML in many different ways, a standard RDF/XML serialisation of BIBFRAME would require the use of an XML schema to ensure that all systems can handle the RDF data as pure XML.[1]

This approach has manifold consequences. On one hand the perception is that the focus in BIBFRAME shifts from an abstract model (based on RDF which is an abstract syntax) to an actual serialisation of the data which implies that the data format is driving the model and not vice versa. Further, the requirement that the data exchange is not RDF, but a specific XML serialisation of RDF imposes a greater burden on implementers, at least on data providers relying on standard RDF frameworks since they might have to implement a custom RDF/XML writer in order to have a serialisation complying to the XML schema.

It would be beneficial for the discussion if LoC and Zepheira could clarify the background to this use case and supply some more information on the exact requirements on the serialisation. The BIBFRAME Primer states that "the BIBFRAME model is defined in RDF" [2]. If the intention is to embrace the RDF model, there should – however – be no restriction on the serialisation.

[1] It is not explicitly stated that the purpose of a required RDF/XML serialization is that systems should be able to handle the BIBFRAME data as pure XML, ignoring RDF semantics. That, however, many of the examples in the authorities paper are not RDF/XML but XML suggests that this is the intention.

[2] P. 15"

In the context of serialization, statements like "To support the ease of BIBFRAME a required serialization of RDF/XML is defined as the basis for interchange" (4.7) and "For BIBFRAME, there likely will not be an XML schema" (4.10.3)" are too important to be overseen in a collection of use cases.

In section 4.8, the objective to "minimize the amount of specialized ontologies" may be read as "one vocabulary to rule them all". We would like to add (and see added to the text) that other vocabularies can be / are to be set into relationships by alignments.

Comments on section 3. "Use Cases"

In general, we would wish to see all examples being consistent, across point papers, the BIBFRAME vocabulary and the the "Demo" section of bibframe.org. This may be a mid-term goal.

Some of the examples in the paper provide two elements describing a type of a resource, e.g. in **3.1**

```
a bf:Monograph ;  
a bf:Work ;
```

But some others provide just one type, e.g. in **3.7**

```
a bf:Monograph .
```

This could be unified.

In use case **3.3**, there are two steps in the text (emphasis by rh/DNB): "In order to do this, she first finds the Work associated with the specific Instance she has in hand. From there, she explores the various relationships to other related Works." The first step "my instance -- which work?" may be too trivial, however, it may be worth considering to add this step explicitly to the process described in BIBFRAME.

Use case **3.6** is interesting, and caused some discussion here. It may be named "Copying and Linking" instead of just "Linking", as it is a mixture of both these important cataloging techniques. We weren't sure why "giving it a new local URI" is mentioned - what is this URI needed for? Is it for creating a Holding Assertion, or for another purpose? In addition, linking from the Instance to the Work may be sufficient, avoiding redundancies, instead of copying the data.

The URIs in **3.6.2** and **3.6.3** should be clarified, as we think it doesn't make sense to use

```
http://local.library.org/examples/shusterman/test001/i1
```

for both the statement "bf:hasInstance" and as a new local URI.

In use case **3.7** we can imagine that automatically created backlinks would help streamlining the process significantly, instead of a purely manual / intellectual intervention.

In addition, we were somewhat astonished to read "BIBFRAME cataloging environment" and "BIBFRAME cataloging interface". To what extent can or will a new cataloging environment and interface be or become a BIBFRAME environment or interface? We are still emphasizing in every presentation and discussion that BIBFRAME will care for the communication part, and that we don't expect BIBFRAME to succeed the MARC format in terms of cataloging formats. As far as we understood, Eric's statement from January is still valid: "I think what people do in the privacy of their own systems is their business. What we should really be focusing on is how to basically exchange the information from one end to the other." Maybe this needs further clarification, beyond the text of the use case paper.

In **3.7.1**, "Stefano searches for the original Work, in English". However, the aspect of the language is not reflected in the SPARQL code. In addition, the code doesn't say that he searches for the original Work, but for any work. Maybe this touches the whole WEM(I) discussion (a search for the original Work may have to filter out all those Works that are translations of, or expressions of the original Work), or maybe the right attribute should be bf:uniformTitle (<http://bibframe.org/vocab/uniformTitle.html>), labeled "Controlled title". As a consequence, in step 4 "Update fork" Stefano may preserve the title "Bluebeard" as a bf:uniformTitle.

Use case **3.8**, "our DNB use case", is somewhat unusual. It doesn't start in the real world as all the others do, it's on the meta and strategy level, and thus the result is maybe not very impressive. We can imagine changing and reworking it into a use case starting with a concrete user need, and resulting from this the definition of an element that sits in the extended area of BIBFRAME. Maybe a local identifier is a candidate for this.

We have recognized that use case **3.9** was presented by Eric to the ALA SAC Committee during ALA Annual, see <http://de.slideshare.net/zepheiraorg/alasubject-em20130631>, slides 21 to 27.

As a suggestion we'd like to note that use case **3.12** may be renamed to "Pseudonym", instead of "Merging and de-dupe". The term "pseudonym" should at least be used somewhere in the text.

Regarding the usage of owl:sameAs, we'd like to suggest umbel:isLike for consideration. It is semantically not as strict as owl:sameAs, and thus doesn't have the implications named. In addition, umbel:isLike should be given subproperties, e.g. "isPseudonymOf", "isRealNameOf", or similar, etc. The same is true for use case **3.13**, with different subproperties.

In use case **3.14** it is excellent to see examples from each one of the four BIBFRAME core classes! We'd expect to see SPARQL code for the search(es) in this use case.

--- EOF ---